

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-341440

(43)Date of publication of application : 22.12.1998

(51)Int.Cl.

H04N 7/32
// H03M 7/36

(21)Application number : 09-149116

(71)Applicant : NIPPON TELEGR & TELEPH CORP <NTT>

(22)Date of filing : 06.06.1997

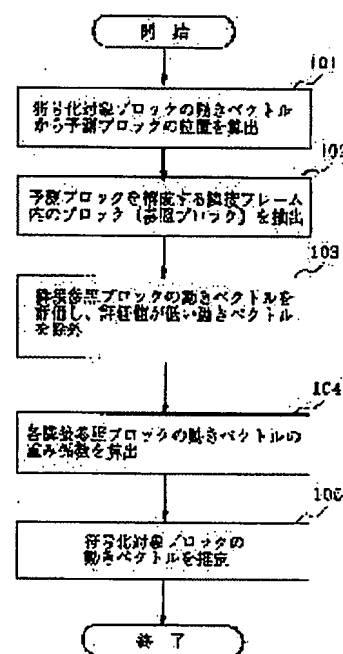
(72)Inventor : SHIMIZU ATSUSHI
KAMIKURA KAZUTO
WATANABE YUTAKA
SAGATA ATSUSHI

(54) MOVING IMAGE ENCODING METHOD AND DEVICE THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce the computing amount of the motion vector detection of inter-frame encoding and the number of times of reference.

SOLUTION: The position of a prediction block is calculated from the motion vector to an adjacent frame of an encoding object block (step 101). One or more reference blocks inside an adjacent reference frame for constituting the prediction block are extracted (step 102). Whether or not the motion vector of the respective adjacent reference blocks indicates an actual motion is evaluated and the motion vector of a low evaluation value is excluded (step 103). The weighting coefficient of the motion vector of the respective adjacent reference blocks is calculated (step 104). The motion vector of the encoding object block is calculated from the motion vector to the adjacent frame of the encoding object block, the motion vector of the respective adjacent reference blocks and the weighting coefficient (step 105).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the
examiner's decision of rejection or application converted
registration]

[Date of final disposal for application]

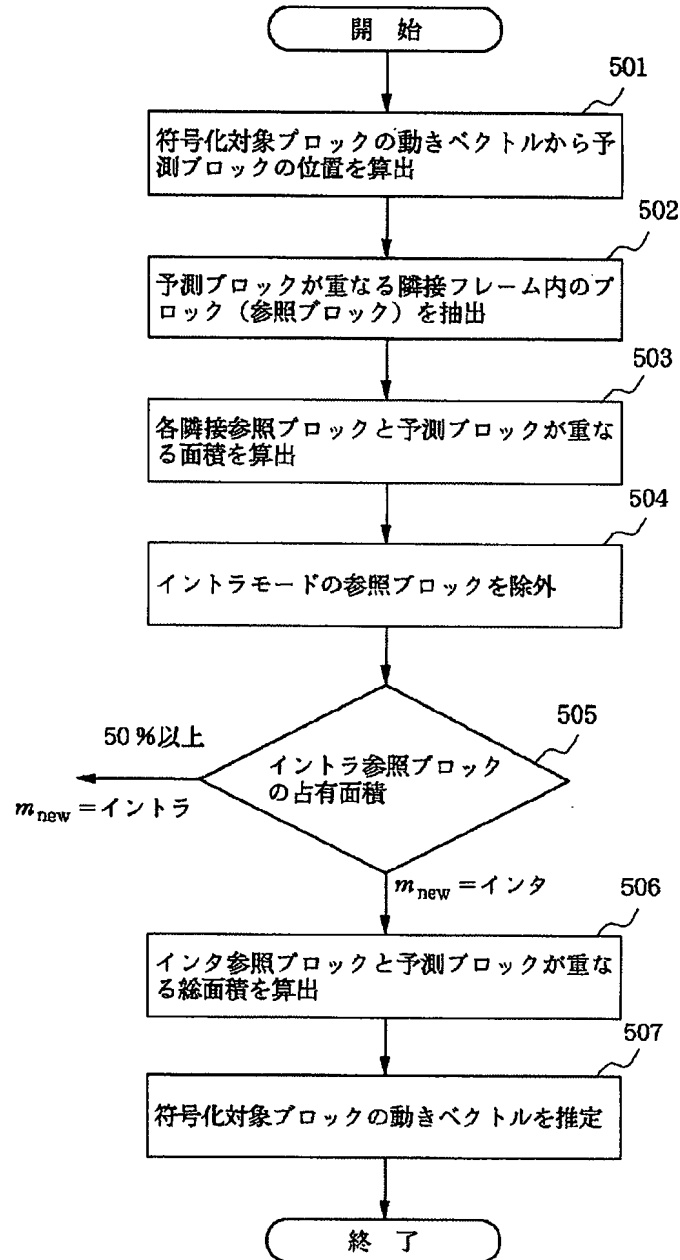
[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of
rejection][Date of requesting appeal against examiner's decision of
rejection]

[Date of extinction of right]

【図8】



フロントページの続き

(72)発明者 嵯峨田 淳
東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-341440

(43) 公開日 平成10年(1998)12月22日

(51) Int.Cl.⁵

識別記号

F I

H 0 4 N 7/32

H 0 4 N 7/137

Z

// H 0 3 M 7/36

H 0 3 M 7/36

審査請求 未請求 請求項の数 8 O L (全 13 頁)

(21) 出願番号 特願平9-149116

(22) 出願日 平成9年(1997)6月6日

(71) 出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72) 発明者 清水 淳

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72) 発明者 上倉 一人

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(72) 発明者 渡辺 裕

東京都新宿区西新宿三丁目19番2号 日本
電信電話株式会社内

(74) 代理人 弁理士 若林 忠

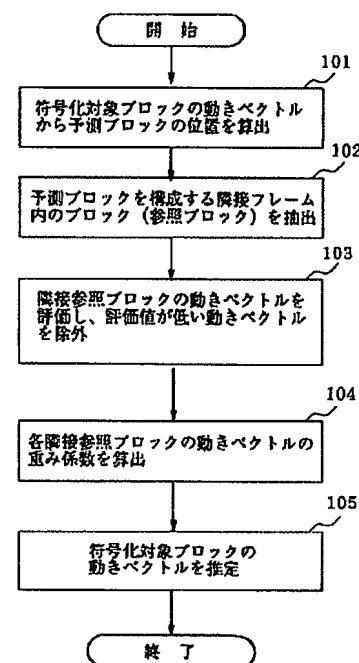
最終頁に続く

(54) 【発明の名称】 動画像符号化方法および装置

(57) 【要約】

【課題】 フレーム間符号化の動きベクトル検出の演算量と参照回数を減らす。

【解決手段】 符号化対象ブロックの隣接フレームへの動きベクトルから予測ブロックの位置を算出する(ステップ101)。予測ブロックを構成する隣接参照フレーム内の1つ以上の参照ブロックを抽出する(ステップ102)。各隣接参照ブロックの動きベクトルが実際の動きを表しているかを評価し、評価値が低い動きベクトルを除外する(ステップ103)。各隣接参照ブロックの動きベクトルの重み係数を算出する(ステップ104)。符号化対象ブロックの隣接フレームへの動きベクトルと、各隣接参照ブロックの動きベクトルと重み係数から符号化対象ブロックの動きベクトルを算出する(ステップ105)。



【特許請求の範囲】

【請求項1】 入力されたフレームをブロックに分割し、ブロック毎に過去または未来の隣接するフレームに対する動きベクトルを検出し、検出された動きベクトルにより動き補償フレーム間予測を行う動画像符号化方法において、符号化対象フレームが隣接するフレーム以外のフレームを参照フレームとする際、該参照フレームと符号化対象フレームの間に存在するブロック毎に検出された動きベクトルから、参照フレームと符号化対象フレーム間の動きベクトルを推定することを特徴とする動画像符号化方法。

【請求項2】 符号化対象フレームの各ブロックの動きベクトルを推定する際、符号化対象ブロックから隣接するフレームへの動きベクトルを検出し、検出された動きベクトルにより求める予測ブロックを構成する隣接フレームのブロックを抽出し、抽出したブロックの情報をもとに、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する、請求項1記載の方法。

【請求項3】 符号化対象フレームと予測ブロックを構成する隣接フレームのブロックが持つ動きベクトルをもとに、符号化対象ブロックの動きベクトルを推定する際、予測ブロックを構成する隣接フレームの各ブロックについて、予測ブロックの構成比率を算出し、算出された構成比率をもとに重み係数を算出し、隣接フレーム内の各ブロックの動きベクトルに重み付けをし、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する、請求項2記載の方法。

【請求項4】 符号化対象フレームと抽出した隣接フレームのブロックが持つ動きベクトルをもとに、符号化対象ブロックの動きベクトルを推定する際、隣接フレームの抽出したブロックの動きベクトルがどの程度実際の動きを表しているかを評価し、該評価値の低い動きベクトルを用いずに、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する、請求項2記載の方法。

【請求項5】 動き補償フレーム間予測を用いる動画像符号化方法において、符号化対象フレームが隣接フレーム以外のフレームを参照する際、参照フレームと符号化対象フレームの間に存在するフレームが複数ある場合、請求項1から4のいずれかに記載の動きベクトル推定を複数回繰り返すことにより、参照フレームから符号化対象フレームへの動きベクトルを推定することを特徴とする動画像符号化方法。

【請求項6】 入力されたフレームをブロックに分割し、ブロック毎に過去または未来の隣接フレームに対する動きベクトルを検出し、検出された動きベクトルにより動き補償フレーム間予測を行う動画像符号化装置において、

符号化対象フレームの各符号化対象ブロックの隣接フレームへの動きベクトルから決まる隣接フレーム内での予測ブロックの位置に存在するブロックが持つ動きベクトルをもとに参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する動きベクトル推定手段を有することを特徴とする動画像符号化装置。

【請求項7】 前記動きベクトル推定手段は、予測ブロックを構成する隣接フレームの各ブロックについて、予測ブロックの構成比率を算出し、算出された構成比率をもとに重み係数を算出し、隣接フレーム内の各ブロックの動きベクトルに重み付けをし、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する請求項6記載の装置。

【請求項8】 前記動きベクトル推定手段は、隣接フレームの抽出したブロックの動きベクトルがどの程度実際の動きを表しているかを評価し、該評価値の低い動きベクトルを用いずに、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する請求項6記載の装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、入力されたフレームをブロックに分割し、ブロック毎に過去または未来の隣接するフレームに対する動きベクトルを検出し、検出された動きベクトルにより動き補償フレーム間予測を行う動画像符号化方法に関する。

【0002】

【従来の技術】一般に、動画像符号化方法では、時間方向の冗長性を除去するため、動き補償フレーム間予測が用いられてる。動き補償フレーム間予測とは、参照フレームを動きベクトルにより動き補償をした画像信号を予測信号とし、符号化対象ブロックの予測誤差を算出する方法である。動き補償フレーム間予測を用いる場合、動きベクトルを探索をする必要がある。

【0003】通常、動きベクトルを探索範囲は、符号化対象フレームと参照フレーム間の距離に比例して拡大される。フレーム間距離が1での動きベクトルの探索範囲を $\pm S \times \pm S$ とする。フレーム間距離が d では、 $\pm d \cdot S \times \pm d \cdot S$ の範囲を探索する。よって、フレーム間距離 d での探索範囲は、フレーム間距離が1の隣接フレームでの動きベクトル探索に対し、 $d \times d$ 倍となる。フレーム間距離1での動きベクトルの探索範囲が $\pm S \times \pm S$ での演算回数を U とすると、フレーム間距離 d での演算回数は $d^2 \cdot U$ となる。

【0004】動きベクトルの探索は膨大な演算回数を伴うため、演算回数を削減する方法が必要とされている。一般に、演算回数を削減する方法として、テレスコピックサーチが用いられる。テレスコピックサーチは、参照フレームと符号化対象フレームの間に存在するフレーム

に対し順次動き探索を行うことにより、動きベクトルを検出する手法である。フレーム間距離 d での演算量は $d \cdot U$ となり、演算回数を削減できる。

【0005】テレスコピックサーチの例を図9に示す。今、フレーム $n+3$ を符号化対象フレーム、フレーム n を参照フレームとして、動きベクトル検出を行う場合を示す。まず、探索範囲 $\pm S \times \pm S$ で、フレーム $n+2$ への動きベクトル v_1 を検出する。続いて、動きベクトル v_1 移動した位置を中心に、探索範囲 $\pm S \times \pm S$ で、フレーム $n+3$ からフレーム $n+1$ への動きベクトル $v_2 + v_1$ を検出する。同様に、動きベクトル $v_2 + v_1$ 移動した位置を中心に探索を行い、フレーム $n+3$ からフレーム n への動きベクトル $v_3 + v_2 + v_1$ を検出する。この動きベクトルを符号化対象ブロックの動きベクトル v_0 とする。この例での演算回数は $3U$ となる。

【0006】このように、テレスコピックサーチを用いることで、演算回数を削減できる。

【0007】

【発明が解決しようとする課題】従来の方法では、符号化対象フレームと参照フレーム間のフレームに対し、順次動き探索を行っている。

【0008】いま、図10に示すような参照関係で符号化処理を行う場合を考える。フレーム間距離1での動きベクトルの探索範囲が $\pm S \times \pm S$ 、その演算回数を U とする。テレスコピックサーチでは、参照関係P1の演算回数は U 、参照関係P2の演算回数は $2U$ 、参照関係P3の演算回数は $3U$ というように、符号化対象フレームと参照フレームとのフレーム間距離に比例して、符号化対象フレームの演算量が増加する問題がある。

【0009】また、参照関係P2とP3では、それぞれ参照関係と無関係なフレームを参照しなければならず、その参照回数はフレーム間距離に比例する。

【0010】このように、テレスコピックサーチでは、各フレームの参照関係が一樣でない場合、参照関係のフレーム間距離に比例して演算量や参照回数が増加し、演算量や各フレームの参照回数が均一にならないという問題がある。このため、例えば、実時間で符号化処理を行う場合、最大演算量と最大参照回数により動画像符号化処理部を設計する必要があった。

【0011】本発明の目的は、フレーム間符号化の動きベクトル検出の演算量と参照回数を削減する動画像符号化方法および装置を提供することにある。

【0012】本発明の他の目的は、符号化対象フレームと参照フレームのフレーム間距離によらずフレーム間符号化の動きベクトル検出の演算量と参照回数を均一にする動画像符号化方法および装置を提供することにある。

【0013】

【課題を解決するための手段】本発明の動画像符号化方法は、符号化対象フレームが隣接するフレーム以外のフレームを参照フレームとする際、該参照フレームと符号

化対象フレームの間に存在するブロック毎に検出された動きベクトルから、参照フレームと符号化対象フレーム間の動きベクトルを推定する。

【0014】本発明の実施態様によれば、符号化対象フレームの各ブロックの動きベクトルを推定する際、符号化対象ブロックから隣接するフレームへの動きベクトルを検出し、検出された動きベクトルにより求まる予測ブロックを構成する隣接フレームのブロックを抽出し、抽出したブロックの情報をもとに、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する。

【0015】本発明の実施態様によれば、符号化対象フレームと予測ブロックを構成する隣接フレームのブロックが持つ動きベクトルをもとに、符号化対象ブロックの動きベクトルを推定する際、予測ブロックを構成する隣接フレームの各ブロックについて、予測ブロックの構成比率を算出し、算出された構成比率をもとに重み係数を算出し、隣接フレーム内の各ブロックの動きベクトルに重み付けをし、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する。

【0016】本発明の実施態様によれば、符号化対象フレームと抽出した隣接フレームのブロックが持つ動きベクトルをもとに、符号化対象フレームの各ブロックの動きベクトルを推定する際、隣接フレームの抽出したブロックの動きベクトルがどの程度実際の動きを表しているかを評価し、該評価値の低い動きベクトルを用いずに参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する。

【0017】本発明の実施態様によれば、動き補償フレーム間予測を用いる動画像符号化方法において、符号化対象フレームが隣接フレーム以外のフレームを参照する際、参照フレームと符号化対象フレームの間に存在するフレームが複数ある場合、上記のいずれかの動きベクトル推定を複数回繰り返すことにより、参照フレームから符号化対象フレームへの動きベクトルを推定する。

【0018】本発明の動画像符号化装置は、符号化対象フレームの各符号化対象ブロックの隣接フレームへの動きベクトルから決まる隣接フレーム内での予測ブロックの位置に存在するブロックが持つ動きベクトルをもとに参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する動きベクトル推定手段を有する。

【0019】本発明の実施態様によれば、動きベクトル推定手段は、予測ブロックを構成する隣接フレームの各ブロックについて、予測ブロックの構成比率を算出し、算出された構成比率をもとに重み係数を算出し、隣接フレーム内の各ブロックの動きベクトルに重み付けをし、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する。

【0020】本発明の実施態様によれば、動きベクトル

推定手段は、隣接フレームの抽出したブロックの動きベクトルがどの程度実際の動きを表しているかを評価し、該評価値の低い動きベクトルを用いずに、参照フレームから符号化対象フレームへの動きベクトルを符号化対象ブロック毎に推定する。

【0021】本発明は、符号化対象フレームとフレーム間距離が1であるフレーム（以下、隣接フレーム）で動きベクトルを検出し、その動きベクトルから、隣接フレーム以外を参照する参照関係の動きベクトルを推定する。

【0022】本発明は、図1に示す手順により符号化対象ブロックから参照フレームへの動きベクトルを推定する。

【0023】1. 符号化対象ブロックの隣接フレームへの動きベクトルから予測ブロックの位置を算出する（ステップ101）。

【0024】2. 予測ブロックを構成する隣接参照フレーム内の1つ以上の参照ブロックを抽出する（ステップ102）。

【0025】3. 各隣接参照ブロックの動きベクトルが実際の動きを表しているかを評価し、評価値が低い動きベクトルを除外する（ステップ103）。

【0026】4. 各隣接参照ブロックの動きベクトルの重み係数を算出する（ステップ104）。

【0027】5. 符号化対象ブロックの隣接フレームへの動きベクトルと、各隣接参照ブロックの動きベクトルと重み係数から符号化対象ブロックの動きベクトルを算出する（ステップ105）。

【0028】次に、本発明の基本構成を、過去のフレームを参照する場合を例にして述べる。本発明の基本構成を図2に示す。まず、入力されたフレームは、ブロック分割部201でブロックに分割される。動きベクトル検出部202で、フレームメモリ203に蓄えられている直前に入力されたフレームからの動きベクトルを検出する。検出された動きベクトルは動きベクトルメモリ204に記憶される。フレームメモリ203内のフレームが参照フレームである場合、スイッチ206を“D”に切り替え、検出した動きベクトルを符号化処理部207に渡し、符号化を行う。フレームメモリ203内のフレームが参照フレームでない場合、動きベクトルメモリ204に記憶されている動きベクトルを用い、動きベクトル推定部205で、符号化対象フレームから参照フレームまでの動きベクトルを推定する。スイッチ206を“E”に切り替え、推定した動きベクトルを符号化処理部207に渡し、符号化を行う。ここで、動きベクトルメモリ204の容量は、符号化対象フレームと参照フレームのフレーム間距離の最大値に比例する。例えば、先に示した図10の参照関係では、2フレーム分の動きベクトルを記憶できるだけの容量が必要となる。

【0029】提案方法の動作例として、図10の参照関

係での動きベクトル検出を考える。既に、フレームnがフレームメモリ203に蓄えられているとする。探索範囲は、 $\pm S \times \pm S$ での1フレーム当りの演算回数をU、ブロック数をNとする。各フレームの参照関係と動きベクトルの関係を図3に示す。

【0030】フレームn+1が入力された場合、隣接フレームnとの間で、符号化対象フレームのブロックi（ $i = 1, 2, \dots, N$ ）の動きベクトル

【0031】

【外1】

$$\vec{V}_{01}(i)$$

を検出する。動きベクトル

【0032】

【外2】

$$\vec{V}_{01}(i)$$

は、動きベクトルメモリ204に記憶される。参照関係P1は、隣接フレーム間の参照なので、スイッチ206を“D”に切り替え、検出した動きベクトルを符号化処理部207に渡し、符号化を行う。

【0033】次に、フレームn+2が入力された場合、隣接フレームn+1との間で、符号化対象フレームのブロック毎に動きベクトル

【0034】

【外3】

$$\vec{V}_{12}(i)$$

を検出する。先程と同様に動きベクトル

【0035】

【外4】

$$\vec{V}_{12}(i)$$

は動きベクトルメモリ204に記憶される。参照関係P2は、隣接フレーム間参照ではないので、動きベクトル

【0036】

【外5】

$$\vec{V}_{02}(i)$$

の推定を行う。動きベクトル推定部205で

【0037】

【外6】

$$\vec{V}_{12}(i)$$

と、動きベクトルメモリ204に記憶されている動きベクトル V_{01} から推定する。

【0038】

【数1】

$$\vec{V}_{02}(i) = f_{02}(V_{01}, \vec{V}_{12}(i)) \dots\dots\dots (1)$$

(5)

特開平10-341440

8

【0039】ここで、 V_{0i} はフレーム n とフレーム $n+1$ 間で検出された全ブロックの動きベクトルである。参照関係 $P2$ は隣接フレーム間参照ではないので、スイッチ 206 を“E”に切り替え、推定した動きベクトル

【0040】

【外7】

$$\vec{v}_{02}(i)$$

を符号化処理部 207 に渡し、符号化を行う。

【0041】次に、フレーム $n+3$ が入力された場合、隣接フレーム $n+2$ との間で、符号化対象フレームのブロック毎に動きベクトル

【0042】

【外8】

$$\vec{v}_{23}(i)$$

を検出する。先程と同様に動きベクトル

【0043】

【外9】

$$\vec{v}_{23}(i)$$

は、動きベクトルメモリ 204 に記憶される。参照関係 $P3$ は、隣接フレーム間参照ではないので、動きベクトル

【0044】

【外10】

$$\vec{v}_{03}(i)$$

の推定を行う。動きベクトル推定部 205 で、

【0045】

【外11】

$$\vec{v}_{23}(i)$$

と、動きベクトルメモリ 204 に記憶されている動きベクトル V_{0i} 、 V_{12} から推定する。

【0046】

【数2】

$$\vec{v}_{03}(i) = f_{03}(V_{01}, V_{12}, \vec{v}_{23}(i)) \dots \dots (2)$$

【0047】ここで、 V_{0i} はフレーム n とフレーム $n+1$ 間で検出された全ブロックの動きベクトル、 V_{12} はフレーム $n+1$ とフレーム $n+2$ 間で検出された全ブロックの動きベクトルである。参照関係 $P3$ は、隣接フレーム間参照ではないので、スイッチ 206 を“E”に切り替え、推定した動きベクトル

$$\vec{v}_{new} = \sum_{i=1}^4 w_i \cdot \vec{v}_i + \vec{v}_0 \quad \left(\sum_{i=1}^4 w_i = 1 \right) \dots \dots (3)$$

【0057】ここで、重み w_i は、隣接参照ブロックと予測ブロックとの関係から算出されるものである。例え

* 【0048】

【外12】

$$\vec{v}_{03}$$

を符号化処理部 207 に渡し、符号化を行う。

【0049】次に、動きベクトル推定部 205 について述べる。符号化対象ブロックと参照フレームの関係を図 4 に示す。この図は、図 3 における参照関係 $P2$ を示している。符号化対象フレームを $n+2$ 、参照フレームを n とする。

【0050】まず、符号化対象ブロックの隣接フレームへの動きベクトル

【0051】

【外13】

$$\vec{v}_0$$

から、隣接フレーム内での予測ブロックの位置を求める。予測ブロックの位置に存在する 1 つ以上のブロック（以下、隣接参照ブロック）を持つ動きベクトル

【0052】

【外14】

$$\vec{v}_i$$

を符号化データから抽出する。隣接参照ブロック数は、ブロック形状や予測ブロックと隣接参照ブロックの配置により決まる。図 4 の例では、ブロック形状が正方形で、予測ブロックは、4 つの隣接参照ブロック (block_i) ($i=1 \sim 4$) と重なり、4 つの動きベクトル

【0053】

【外15】

$$\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$$

が存在する。この隣接参照ブロックの動きベクトル

【0054】

【外16】

$$\vec{v}_i$$

に対して重み付けし、次式より新たな動きベクトル

【0055】

【外17】

$$\vec{v}_{new}$$

【0056】を算出する。

【数3】

ば、予測ブロックに示す各隣接参照ブロック block_i の面積比率から算出することができる。この他、符号化対象

ブロックの画像信号と各隣接参照ブロックの画像信号の相関係数などがある。また、隣接参照ブロックの動きベクトルの信頼度を算出し、信頼度の低い動きベクトルを除外して推定することにより、動き補償フレーム間予測効率を向上させることができる。動きベクトルの信頼度としては、動き検出時の予測誤差や符号化モード、動きベクトルの向きや大きさがある。

【0058】次に、符号化対象フレームと参照フレームのフレーム間距離が3以上の場合について述べる。参照関係P3の場合、まず、フレームn+3からフレームn+1への動きベクトル

【0059】

【外18】

$$\vec{V}_{13}$$

を上記の方法で推定する。そして、その動きベクトル

【0060】

【外19】

$$\vec{V}_{13}$$

を図4の

【0061】

【外20】

$$\vec{V}_0$$

とし、フレームn+1の動きベクトル V_n をもとに、動きベクトル

【0062】

【外21】

$$\vec{V}_{03}$$

を推定する。フレーム間隔が3以上の場合でも同様に、繰り返し推定することで、符号化対象フレームと参照フレームの間の動きベクトルを推定することができる。

【0063】未来方向の予測は、上記の参照の順番を逆にすることで容易に実現できる。

【0064】提案方法での演算量を、以下に示す。まず、隣接フレーム間での動きベクトル探索回数は隣接フレーム間の探索のみで、 V_{01} 、 V_{12} 、 V_{23} の3回である。よって、演算量は3Uとなる。また、各フレーム(n, n+1, n+2)の参照回数はそれぞれ1回である。

【0065】参考までに、同様の参照関係をテレスコピックサーチで動きベクトル検出を行った場合の演算量と参照回数を示すと、演算量は $U + (U+U) + (U+U+U) = 6U$ となり、各フレーム(n, n+1, n+2)の参照回数はそれぞれ3回、2回、1回である。

【0066】

【発明の実施の形態】次に、本発明の実施の形態について図面を参照して説明する。

【0067】図1は本発明の一実施形態の動画画像符号化装置の構成図である。本実施形態の符号化アルゴリズムは、動き補償フレーム間予測と離散コサイン変換(DCT)を基本としている。動き補償フレーム間予測は、先に示した図10の参照関係のうち、参照関係P1とP2で行う。ブロック形状は正方形とし、動きベクトルを推定するための重み係数は面積比率から算出する。本実施形態では、符号化モードは、フレーム間符号化モード

(以下、インタモード)とフレーム内符号化モード(以下、イントラモード)の2種類がある。動きベクトル検出時の予測誤差の大きさにより符号化モードをイントラモードに切り替えている。

【0068】図6中、図2と同じ参照番号は同じ構成要素を示している。

【0069】符号化処理部207はブロック分割部301とフレームメモリ302と動き補償部303とスイッチ304と減算器305とDCT部306と量子化部307と逆量子化部308とIDCT部309と加算器310と符号化部311で構成されている。

20 【0070】まず、入力されたフレームは、ブロック分割部201でブロックに分割される。動きベクトル検出部202で、フレームメモリ203に蓄えられている直前に入力されたフレームからの動きベクトル v_0 を検出する。その際、予測誤差 E_{rr} から、符号化対象ブロックの符号化モード m_0 を次式により決定する。

【0071】

【数4】

$$m_0 = \begin{cases} \text{Intra} & E_{rr} \geq TH \\ \text{Inter} & E_{rr} < TH \end{cases} \quad \text{----- (3)}$$

30

【0072】ここで、THは、符号化モード決定のための閾値である。検出した動きベクトル v_0 を動きベクトルメモリ204に、符号化モード m_0 を符号化モードメモリ208に記憶する。参照関係がP1のように、フレームメモリ203内のフレームが参照フレームである場合、スイッチ206Aおよびスイッチ206Bを“D”に切り替え、検出した動きベクトルと符号化モード m_0 を符号化処理部207に渡し、符号化処理を行う。また、参照関係がP2のように、フレームメモリ203内のフレームが参照フレーム以外の場合、動きベクトルメモリ204に記憶されている動きベクトルと、符号化モードメモリ208に記憶されている符号化モードを用い、動きベクトル推定部205で、符号化対象フレームから参照フレームまでの動きベクトルを推定する。スイッチ206Aおよびスイッチ206Bを“E”に切り替え、推定した動きベクトル

【0073】

【外22】

$$\vec{V}_{new}$$

50

と符号化モード m_{new} を符号化処理部207に渡し、符号化処理を行う。

【0074】符号化処理部207では、符号化対象フレームをブロックに分割し、先に決定された符号化モードに基づき、符号化処理を行う。符号化モードがインタである場合は、先に算出された動きベクトルにより、フレームメモリ302内の符号化済みフレームに対し動き補償部303で動き補償を行い、予測ブロックを抽出する。符号化モードがイントラの場合、予測ブロックの信号は0とする。予測ブロックと符号化対象ブロックの差分にDCT部306でDCTを施し、量子化部307で量子化した後、符号化部311で可変長符号化を行う。また、量子化された信号は、逆量子化部308で逆量子化した後、IDCT部309で逆DCTを施し、加算器310で予測ブロックの信号を加えて、復号する。この復号信号はフレームメモリ302に蓄えられ、次に符号化対象フレームの動き補償に用いられる。

【0075】図7は符号化対象ブロックの符号化モード決定手順を示す流れ図である。

【0076】1. 符号化対象ブロックの隣接フレームへの動きベクトル

【0077】

【外23】

\vec{v}_0

と符号化モード m_0 を算出する(ステップ401)。

【0078】2. ステップ402で、符号化対象ブロックの符号化モード m_0 がイントラモードである場合は、フレーム内符号化にて符号化を行う。

【0079】3. ステップ403で、参照フレームが隣接フレームである場合は、検出した動きベクトル

【0080】

【外24】

\vec{v}_0

を動き補償フレーム間予測に用いる。

【0081】4. 隣接フレームの各参照ブロックの符号化モードと動きベクトルから、参照フレームへの動きベクトル

【0082】

【外25】

\vec{v}_{new}

と符号化モード m_{new} を算出する(ステップ404)。

【0083】5. 算出した符号化モード m_{new} により、符号化対象ブロックの符号化モードを決定する。

【0084】上記の手順4における動きベクトル

【0085】

【外26】

\vec{v}_{new}

と符号化モード m_{new} の算出方法は、以下の通りである。図8に、動きベクトル算出方法の流れ図を示す。

【0086】1. 符号化対象ブロックの動きベクトル

【0087】

【外27】

\vec{v}_0

から予測ブロックの位置を算出する(ステップ501)。

【0088】2. 予測ブロックが重なる隣接フレーム内の1つ以上の参照ブロックを抽出する(ステップ502)。予測ブロックの位置により1~4つの隣接参照ブロック $block_i$ と重なる可能性がある(図5)参照ブロック数をNBとする。

【0089】3. 予測ブロックと各隣接参照ブロック $block_i$ ($i=1, \dots, NB$) が重なり合う面積 p_i を算出する(ステップ503)。

【0090】4. 各隣接参照ブロック $block_i$ の符号化モード m_i を調べ、イントラモードの参照ブロックを除外する(ステップ504)。

【0091】5. ステップ505で、イントラモードの隣接ブロックの占有面積が50%を越える場合、符号化対象ブロックの符号化モード m_{new} をイントラモードとする。

【0092】6. インタモードの隣接ブロック $Inter_block_i$ ($i=1, \dots, NIB \leq NB$) が占める総面積 p_{all} を計算する(ステップ506)。ここで、NIBはインタモード隣接ブロックの数、 $Inter_p_i$ は、各インタモード隣接ブロックの占有面積である。

【0093】

【数5】

$$p_{all} = \sum_{i=1}^{NIB} Inter_p_i$$

7. 各インタモード隣接ブロックの占有面積 $Inter_p_i$ 、各隣接ブロックの動きベクトル

【0094】

【外28】

\vec{v}_i

符号化対象ブロックの隣接フレームへの動きベクトル

【0095】

【外29】

\vec{v}_0

から、符号化対象ブロックの動きベクトル

【0096】

【外30】

$$\vec{v}_{new}^{13}$$

を算出する。

【0097】

【数6】

$$\vec{v}_{new} = \frac{1}{p_{all}} \sum_{i=1}^{NIB} Inter - p_i \cdot \vec{v}_i + \vec{v}_0$$

【0098】ここで、手順4でイントラモードのブロックを除外している。符号化モードがイントラモードであることは、動きベクトル検出時の予測誤差が大きいことを示しており、その動きベクトルの信頼性は低いと考えられる。そこで、信頼性が低い動きベクトルを除外するため、イントラモードを除外している。

【0099】以上述べた実施形態では、重み係数として面積比率を用いたが、符号化対象ブロック画像信号と参照ブロックの画像信号間の相関係数でもよい。動きベクトルの正しさの評価値として符号化モードを用いたが、動きベクトル検出時の予測誤差や動きベクトルの向きでもよい。また、ブロックを正方形としたが、これにと

【0100】

【発明の効果】以上説明したように、本発明によれば、動きベクトル探索の演算量とフレームメモリの参照回数を減らすことができ、かつ、演算量と参照回数は符号化対象フレームと参照フレームのフレーム間距離によらない。結果として、演算量と参照回数を低減し、演算量と参照回数が様な動きベクトル探索が可能となる。

【図面の簡単な説明】

【図1】本発明の動きベクトル推定方法の流れ図である。

【図2】本発明の動画画像符号化装置のブロック図である。

【図3】動きベクトルの推定を示す図である。

* 【図4】符号化対象ブロックと参照フレームの関係を示す図である。

【図5】予測ブロックと重なる参照ブロックを示す図である。

【図6】本発明の一実施形態の動画画像符号化装置のブロック図である。

【図7】図6の実施形態における、符号化対象ブロックの符号化モード決定手順を示す流れ図である。

【図8】図6の実施形態における動きベクトル算出の流れ図である。

【図9】テレスコピックサーチを示す図である。

【図10】フレーム間の参照関係を示す図である。

【符号の説明】

101～105 ステップ

201 ブロック分割部

202 動きベクトル検出部

203 フレームメモリ

204 動きベクトルメモリ

205 動きベクトル推定部

206, 206A, 206B スイッチ

207 符号化処理部

301 ブロック分割部

302 フレームメモリ

303 動き補償部

304 スイッチ

305 減算器

306 DCT部

307 量子化部

308 逆量子化部

309 IDCT部

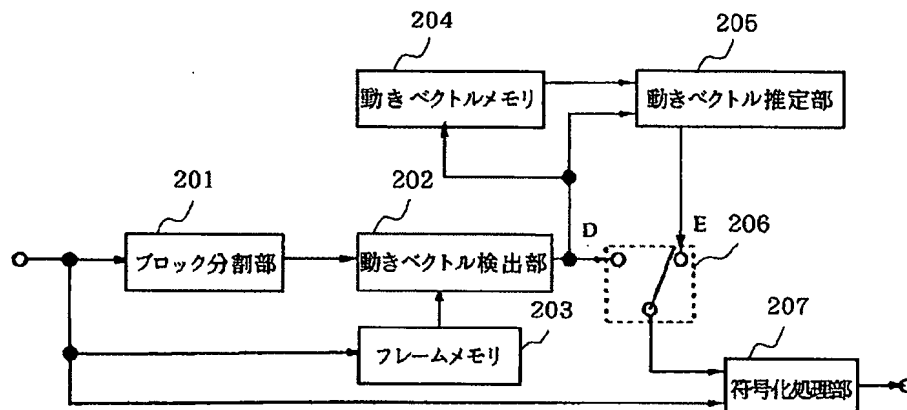
310 加算器

311 符号化部

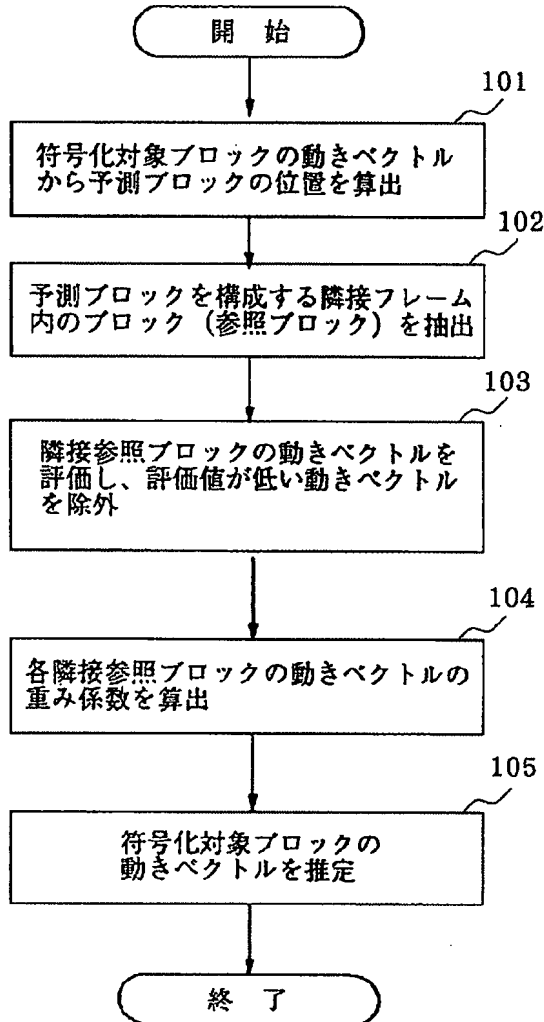
401～405 ステップ

* 501～507 ステップ

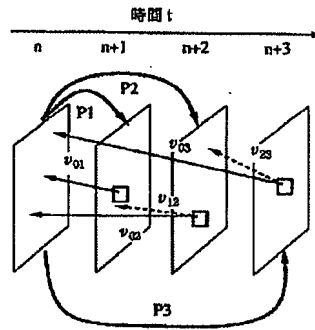
【図2】



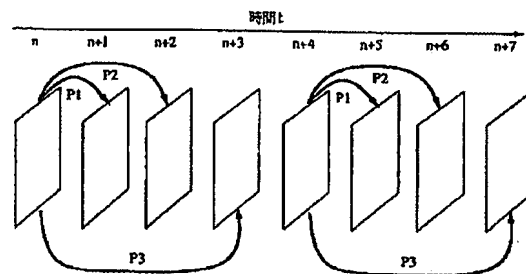
【図1】



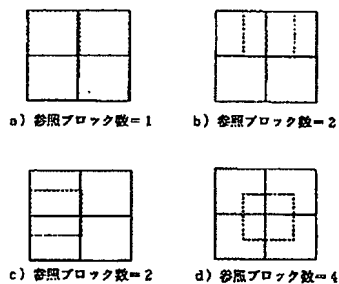
【図3】



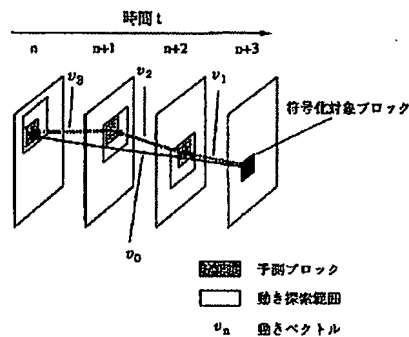
【図10】



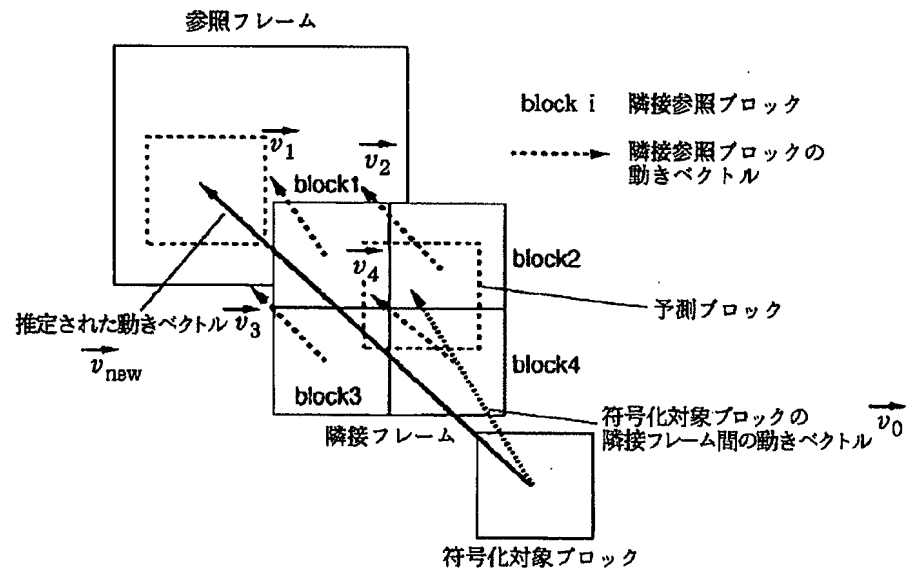
【図5】



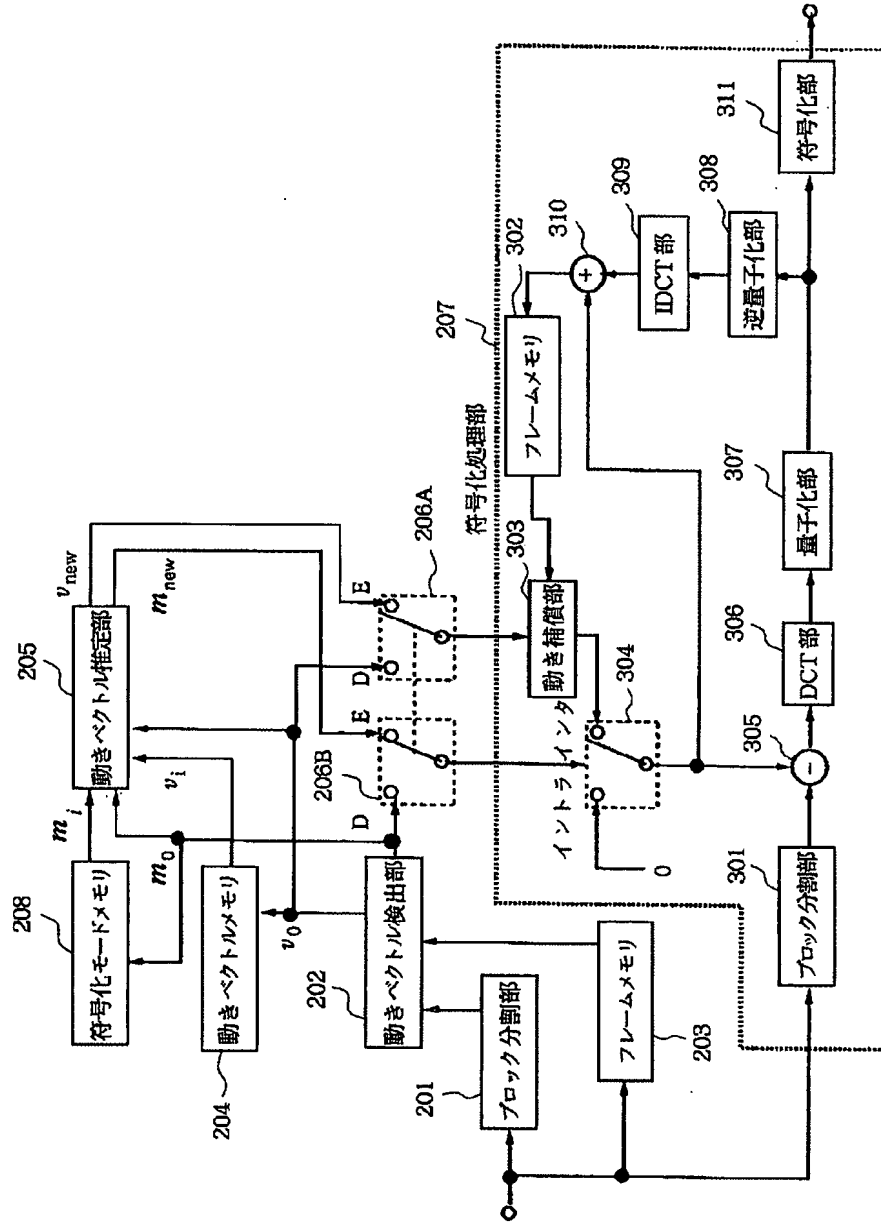
【図9】



【図4】



【图 6】



【図7】

